

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **David Kneys**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: GIRITON Systems s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Gaura**

Konzultant bakalářské práce: Ing. Jan Gřeš, MSc.

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2015

.....*Oknejs*.....

Rád bych poděkoval svému konzultantovi Janu Gřešovi, MSc. za umožnění a příkladné vedení v absolvování praxe ve firmě GIRITON Systems s.r.o., svým kolegům, se kterými jsme měli možnost spolupracovat a vyměňovat si zkušenosti a také panu Ing. Janu Gaurovi za vedení při tvorbě bakalářské práce.

Abstrakt

Obsahem této bakalářské práce je průběh mého působení ve firmě GIRITON Systems s.r.o. v rámci odborné praxe. Nejprve zde informuji o zaměření a oblasti odborného působení firmy, dále se zabývám technologiemi a nástroji, se kterými jsem během své praxe pracoval a také uvádím příklady úkolů, na kterých jsem se podílel. V závěru práce hodnotím celkový přínos absolvované praxe.

Klíčová slova: bakalářská práce, odborná praxe, Java, Android, Vaadin, JasperReports

Abstract

The content of this thesis is my work for company GIRITON Systems s.r.o. in the context of professional practice. First I inform about the focus and area of expertise of the company. Next I describe technologies and tools, which I worked with. Finally I publish examples of tasks, which I worked on. In the end of the thesis I evaluate overall benefit of the internship.

Keywords: bachelor thesis, professional practice, Java, Android, Vaadin, JasperReports

Seznam použitých zkratk a symbolů

API	– Application Programming Interface
CSS	– Cascading Style Sheets
DOC	– Microsoft Word Document
ERP	– Enterprise Resource Planning
GUI	– Graphical User Interface
GWT	– Google Web Toolkit
HTML	– Hyper Text Markup Language
ORM	– Object Relation Mapping
PDF	– Portable Document Format
POM	– Project Object Model
RMI	– Remote Method Invocation
SVN	– Subversion
XML	– Extensible Markup Language
XLS	– Microsoft Excel Spreadsheet

Obsah

1	Úvod	4
2	Základní informace	5
2.1	Odborné zaměření firmy	5
2.2	Pracovní zařazení studenta	5
3	Seznam úkolů	6
3.1	Android aplikace Píchací hodiny	6
3.2	Android aplikace Osobní docházka	6
3.3	Webová aplikace Docházka	7
3.4	JasperReports	7
4	Postup řešení zadaných úkolů	8
4.1	Android aplikace Píchací hodiny	8
4.2	Android aplikace Osobní docházka	10
4.3	Webová aplikace Docházka	12
4.4	JasperReports	14
5	Závěr	15
5.1	Dosavadní znalosti a jejich aplikace	15
5.2	Nabyté znalosti a zkušenosti během praxe	15
5.3	Celkové zhodnocení	16
6	Reference	17

Seznam obrázků

1	Android aplikace Píchací hodiny	8
2	Android aplikace Osobní docházka	10
3	Aplikace Docházka GIRITON	12

Seznam výpisů zdrojového kódu

1	Jednoduchá Android aplikace	9
2	Vaadin - společný kód pro client a server	13

1 Úvod

Svou bakalářskou práci jsem se rozhodl vypracovat formou bakalářské praxe, protože si myslím, že tato praxe bude velkým přínosem do mého profesního života. Motivací pro mě bylo pracovat na reálném projektu, který má reálné využití. Dalším důvodem proč jsem zvolil tuto cestu je také fakt, že ve firmě působím již skoro dva roky.

Tato bakalářská práce popisuje mou práci v rámci individuální odborné praxe ve firmě GIRITON Systems s.r.o. V práci popisuji zaměření firmy a používané technologie ve firmě. Další kapitola popisuje mé pracovní zařazení společně s pracovní náplní. Dále jsou zde rozebrány vybrané úkoly a úlohy, které jsem měl za úkol vyřešit. U jednotlivých úkolů nejdříve nastíním zadání, dále odhadnu strávený čas na nich a nakonec popisuji svou práci na úkolech se zdůrazněním problémů, které mě při řešení potkaly a také jejich vyřešení.

Ke konci práce se pak nacházejí nejdůležitější části práce jakými jsou zhodnocení dovedností a znalostí týkajících se mé pracovní náplně. Hodnotím zde své dosavadní znalosti, které jsem získal prostřednictvím školy před odbornou praxí s důrazem na jejich využití. Dále zde hodnotím vědomosti a znalosti nabyté z působení ve firmě v průběhu bakalářské praxe. Na závěr shrnuji celkový osobní přínos odborné praxe.

2 Základní informace

Firma GIRITON Systems s.r.o. je zapsána v obchodním rejstříku od 15.3.2011. Pracoviště, které jsem navštěvoval, se nachází v Podnikatelském inkubátoru Vysoké školy báňské - Technické univerzity v Ostravě.

2.1 Odborné zaměření firmy

GIRITON Systems s.r.o. je mladá dynamicky se rozvíjející softwarová firma, která se soustředí na vývoj nejrozličnějších softwarových řešení. Firma vznikla na popud bakalářské práce pana Jana Gřeše, MSc., ze které vznikl úspěšný softwarový produkt pro řízení firemních procesů v malých a středních firmách zaměřených nejen na výrobu. Ze samého počátku se tedy firma orientovala hlavně na ERP systémy a sběr dat, které měly za úkol pomáhat jednotlivým podnikům efektivněji plánovat výrobu.

Dnes se firma soustředí nejen na ERP systémy, ale hlavně na vývoj a distribuci kompletního docházkového systému Docházka GIRITON. Dále se firma zabývá nejrozličnějšími systémy na míru. Všechny tyto systémy jsou vyvíjeny pomocí dnes nejmodernějších technologií postavených převážně na platformě Java.

2.2 Pracovní zařazení studenta

Ve firmě jsme byli zařazeni na pracovní pozici Java programátor, kde mou pracovní náplní byl převážně vývoj a údržba Android [5] aplikací pro systém Docházka GIRITON. Dále jsem se v práci podílel na vývoji cloudové aplikace Docházka GIRITON [14], která je naprogramována v programovacím jazyce Java [1] pomocí frameworku Vaadin [3] a posledním větším úkolem, se kterým jsem se setkal, byla práce s knihovnou JasperReports [11], která pomáhá v generování nejrozličnějších reportů do různých formátů.

3 Seznam úkolů

V kapitole jsou vypsány vybrané úkoly, na kterých jsem se alespoň většinově podílel. Odhad času stráveném na úkolu popisuje čistě můj strávený čas bez ohledu na celkový čas stráveném na projektu.

3.1 Android aplikace Píchací hodiny

V průběhu praxe jsem měl za úkol pracovat na klientské Androidí aplikaci Píchací hodiny, skrze kterou budou uživatelé vykazovat svou docházku. [16]

3.1.1 Zadání

Aplikace může být nainstalována na jakémkoliv Android zařízení se systémem ve verzi Android 4.0.3, které je typicky umístěno někde při vstupu na pracoviště. Aplikace bude sloužit jako terminál a jako jedna z primárních možností jak vložit záznam docházky do systému. První verze aplikace má zvládat jednoduché vkládání docházky ve tvaru např. "pauza start 11:10, pauza stop 11:40". Aplikace musí spolehlivě fungovat jak v offline režimu bez internetového připojení, tak v online režimu s internetovým připojením.

V průběhu vývoje aplikace přibývaly nové a nové požadavky jako jsou například focení obličeje pomocí kamery zařízení, rozšíření možnosti vstupu o práci se čtečkou RFID či NFC čipů nebo také sledování aktuální GPS pozice právě vloženého záznamu.

3.1.2 Časová náročnost

Celkově jsem strávil na aplikaci odhadem 150 hodin, ve kterých je zahrnuto rozšíření mých znalostí o specifické vlastnosti systému Android, několik iterací aplikace, její testování a opravování chyb a především problémy spojené s pochopením a konečné správné implementaci kamery.

3.2 Android aplikace Osobní docházka

Dalším úkolem bylo vytvořit aplikaci, která bude fungovat podobně jako Android aplikace Píchací hodiny, viz kapitola 3.1, ale bude právě právě pro jednoho člověka. [15]

3.2.1 Zadání

Aplikace je určena pro jednoho uživatele, který skrze ní bude vykazovat svou docházku. Dále si bude moci svou vloženou docházku zpětně prohlížet.

Aplikace bude zvládat vše, co Android aplikace Píchací hodiny a navíc bude rozšířena o celkový přehled a podle oprávnění uživatele o administraci již vložených dat. V případě speciálního admin oprávnění může uživatel prohlížet či upravovat docházku všech uživatelů.

3.2.2 Časová náročnost

Na této aplikaci jsem pracoval odhadem 80 hodin a to i přes to, že aplikace byla srovnatelná, či dokonce složitější na implementaci než dříve popisovaná Android aplikace Píchnací hodiny. To především proto, protože se z velké části daly využít již jednou napsané a otestované algoritmy z předchozí Android aplikace a samozřejmě využít nabyté znalosti Android frameworku.

3.3 Webová aplikace Docházka

Dalším úkolem, který probíhal paralelně s ostatními bylo podílení se na vývoji webové aplikace postavené na frameworku Vaadin. [14]

3.3.1 Zadání

Mými úkoly byly různé implementace grafického rozhraní, či složitější a zodpovědnější práce s doménovým modelem, při kterém jsem měl možnost pracovat s doménovou vrstvou aplikace, či implementace různých vzdálených metod, které jsou využity u již zmínovaných Android klientů.

3.3.2 Časová náročnost

Odhadovaný strávený čas na aplikaci je 100 hodin. Tento čas zahrnuje práci na GUI aplikace. Dále implementaci a údržbu větších částí systému doménové logiky včetně její návržení a implementace. Úkolem byla taky implementace metod pro vzdálené volání přes Java RMI. [2]

3.4 JasperReports

Jedním z posledních větších úkolů bylo zavedení reportů různých dat z aplikace pomocí knihovny JasperReports. [11]

3.4.1 Zadání

Připravte data a naimplementujte způsob pro generování reportů ve formátu pdf včetně šablon jednotlivých reportů pomocí knihovny JasperReports.

3.4.2 Časová náročnost

Práce zahrnovala implementaci různých předloh pro reporty, připravení jejich dat a spojení se systémem. To vše zabralo přibližně 70 hodin i s naučením a pochopením samotné knihovny JasperReports.

4 Postup řešení zadaných úkolů

U každého úkolu popisuji svou práci na něm s popisem větších problémů či zajímavostmi, se kterými jsem se při řešení úkolů setkal.

Při řešení úkolů jsem aplikoval buď již nabyté znalosti nebo jsem dodatečně nastudoval materiály pro jejich vyřešení. Většina studované literatury je psána v anglickém jazyce, takže bylo nezbytné ovládat anglický jazyk alespoň na úrovni nezbytné pro čtení a pochopení anglicky psané dokumentace.

4.1 Android aplikace Píchací hodiny

Mým úkolem bylo společně s kolegou naprogramovat celou klientskou část aplikace a později taky její serverovou část. Oba jsme se na tomto projektu Android [5] učili. Proto při vývoji došlo k mnoha přešlapům a slepým cestám. Vycházeli jsme z firemní aplikace dříve napsané pro zcela jiné účely. Z předlohy jsme si vzali hlavně strukturu aplikace, která však byla naprogramována ještě v době, kdy byl Android framework na začátcích svého vývoje. Vycházelo se z toho, že se aplikace skládala z jediné Activity a X TopPanelů. TopPanel byla naše třída, která nejčastěji rozšiřovala základní view komponentu Android LinearLayout. Tento TopPanel představoval vždy jednu obrazovku a obsahoval veškerý kód pro interakci s komponenty na této obrazovce. Zpočátku byl tento způsob architektury dostačující a funkční, nicméně při růstu a rozšiřování aplikace o nové funkcionality se tento způsob začal jevit jako nedostatečný. Nebyl totiž schopný spolehlivě zachytit specifický životní cyklus všech Android aplikací [6]. Příkladem může být třeba implementace kamery, o které níže píšu blíže, kterou si je potřeba od systému správně nejen vyžádat, ale taky včas správně uvolnit pro jiné cizí aplikace. Dále jsme narazili na potřebu aplikace reagovat na stisknutí tlačítka zpět. Na první pohled i tak jednoduchá věc jako je při stisknutí tlačítka zpět vrátit předchozí obrazovku nám značně zkomplikovala život a díky použití nevhodné architektury kód narůstal a stával se čím dál tím více neudržitelným.



Obrázek 1: Android aplikace Píchací hodiny

Nakonec došlo ke klíčovému rozhodnutí aplikaci přepsat tak, aby odpovídala dnešním doporučeným postupům Androidu. [5] Díky nabytým zkušenostem z vývoje jiné aplikace nebyl tento úkol z technického hlediska obtížný, nicméně bylo třeba aplikaci zkontrolovat a ověřit zda funguje vše, co doposud fungovalo a to i přesto, že je aplikace pokryta automatickými testy, protože zásah to byl v konečném důsledku velmi velký. Přepsáním aplikace jsme vyřešili plno dosavadních problémů a aplikace je teď hlavně snadno rozšiřitelná o novou funkcionalitu.

4.1.1 Android framework

Pro programování nativních Android aplikací se využívá Android framework, který obsahuje nástroje a komponenty od práce s GUI aplikace až po API pro přístup k hardwarovým částem zařízení jakými jsou např. reproduktory, různé notifikační diody, proximity sensory, či kamery. Android framework je robustní balík knihovnem, které jsou napsány s ohledem na potřeby a požadavky mobilních zařízení.

```
public class MyActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState){
        setContentView(R.layout.my_layout); //layout defined in xml

        Button btn = (Button)findViewById(R.id.btnMy); //find btn by id setted in xml layout
        btn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){
                //do something
            }
        });
    }
}
```

Výpis 1: Jednoduchá Android aplikace

Použitím Android frameworku nevzniká žádné omezení a v drtivé většině případů je možné použít vše, co obsahuje samotná Java SE. Při vývoji je třeba mít na paměti, že mobilní zařízení mají mnohdy omezené zdroje jako je například baterie zařízení nebo menší výkon a proto taky dávat přednost knihovnám vycházejících přímo z potřeb mobilních zařízení a Androidu.

4.1.2 Problémy

Při vyvíjení aplikace vznikl požadavek na jednoznačnou identifikaci uživatele, který vkládá svou docházku přes terminál. V reálném nasazení se ukázalo, že někteří zaměstnanci používající systém pro vkládání docházky podváděli. V aplikaci nebyl žádný prvek, který by dokázal zkontrolovat, že vkládanou docházku A za uživatele B vložil opravdu uživatel B. Proto se muselo přijít s řešením, které by tento problém omezilo, či úplně odstranilo.

Prvotním nápadem bylo použití terminálů se čtečkou otisků prstů, nicméně tenhle způsob řešení byl brzo zavrhnut z důvodu závislosti na terminálech, které toto zvládají.

S tím také souviselo omezení potenciálně použitelných terminálů a také několikanásobně vyšší náklady na pořízení takového zařízení. Řešení přitom bylo jednoduché. Při vkládání záznamu stačilo pořídit fotografii z přední kamery zařízení, která se dnes v nachází drtivě většině mobilních zařízení. Tyto fotografie se ukládají do systému a v případě podezření na pokus o podvod je možné záznamy zpětně ověřit.

Samotná implementace zabrala hodně času a to především z důvodu nesprávného pochopení Android API vystavěného kolem používání hardwarových kamer. Toto API je totiž celkem nízkoúrovňové a nechává velkou část zodpovědnosti ošetřit mnoho různých nástrah a překážek na programátorovi. V současné verzi Androidu 5 Lollipop je toto API nahrazeno jiným, inteligentnějším a přímočarejším. Velkým překvapením bylo, že různé implementace kamery fungovaly různě na různých zařízeních a to i v případě dvou kusech hardware se stejným modelovým označením. V posledních implementacích se již podařilo kameru naimplementovat správně a odladit všechny chyby. Množina různých zařízení a hlavně různých verzí Androidů, které musí být podporovány je však velká a tak je nemožné říci, že tato implementace funguje 100% na všech zařízeních. Vždy se může objevit zařízení, které se bude muset individuálně řešit.

4.2 Android aplikace Osobní docházka

Mým úkolem bylo naprogramovat obdobně jako u Android Píhací hodiny celou aplikaci, viz kapitola 4.1. Na úkolu jsem pracoval s kolegou, kde jsme si sami rozdělovali jednotlivé části aplikace a práci na ní. Má práce spočívala v implementaci různých obrazovek, persistencí dat a implementací RMI [2] metod volaných aplikací.



Obrázek 2: Android aplikace Osobní docházka

Aplikace byla mnohem složitější než Aplikace android Píhací hodiny, a to hlavně impementací složitějších grafických layoutů, což souvisí s náročnější prací se životním cyklem Android aplikací [6]. Dalším požadavkem bylo mít aplikaci kompatibilní až do verze Android Gingerbread s číselným označením 2.3.x [7]. Doposud jsme programovali

pro mnohem novější verzi Android Ice Cream Sandwich 4.0.x [8] a tak se tento úkol mohl zdát obtížnější z důvodu nutnosti pracovat se starou verzí API.

Android má však výborně zpracovanou podporu starších verzí a zároveň možnost s minimálním omezením používat API a postupy z nejnovějších verzí i ve verzích starších. Toto je možné díky tomu, že s každou novou verzí Androidu přichází zároveň i Android support library [9], která je schopna většinu inovací zpětně dodat.

Postupoval jsem takovým způsobem, že jsem nejdřív naprogramoval nějaký grafický skelet, ve kterém jsem pak kladl důraz především na funkcionalitu aplikace. Samotný vzhled aplikace a doladování grafické části jsem dodělával až na úplný konec. Výsledkem je dobře vypadající aplikace, která spolehlivě funguje i na starších verzí Android.

4.2.1 Motivace

Motivací pro vytvoření aplikace, která by uměla všechno, co Android docházka, ale byla určena jen pro jednoho člověka, byla možnost ovládat systém pouze pomocí chytrého telefonu a to kdykoliv a kdekoliv i bez přístupu k internetu. Zároveň umožňuje oprávněným osobám prohlížet docházku jiných lidí bez nutnosti jakkoliv manuálně interagovat s webovou aplikací. Dále byl touto aplikací celý systém rozšířen o jinou možnost vkládání záznamů a tím se taky zvětšil okruh potenciálních zákazníků celého produktu. Chytrý mobilní telefon má dnes již většina lidí a tak firmám nečiní velký problém tento docházkový systém v rychlém čase nasadit.

4.2.2 Debug

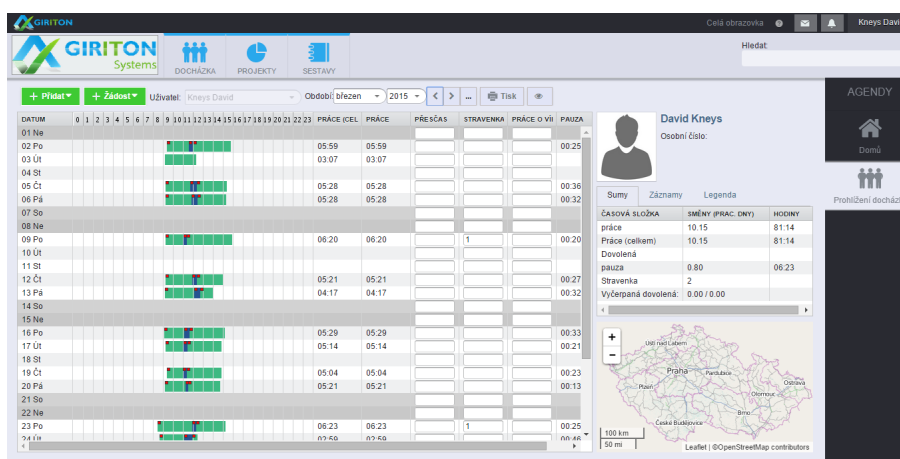
Zajímavostí může být reportování chyb rovnou z aplikace prostřednictvím internetu, na kterém jsem se taktéž svou prací podílel. Tento systém reportování chyb má firma zaveden do všech nejen Android aplikací. Díky tomuto firma ví o chybě v produkci mnohem dříve, než ji stačí uživatel nahlásit. A to i s detaily, které jsou nezbytné pro vyřešení chyby.

Aplikace jsou naprogramovány tak, aby každou chybu logovaly a aby se zároveň zamezilo probublávání vyjímek až na povrch vedoucím k pádům celé aplikace. Aplikace musí alespoň základně fungovat a není důvod, proč by měla celá aplikace selhat jen proto, že se nepodařilo tlačítku správně nastavit barvu.

Tyto logy se v případě aktuálního připojení k internetu rovnou pošlou do serverové aplikace, kde jsou později dostupné k analýze. V případě, že se chyba stala v době, kdy nebylo dostupné připojení k internetu, pak se tento log zapíše do souboru na lokálním zařízení a při nejbližším připojení k internetu pošle na server.

4.3 Webová aplikace Docházka

Produktem, na který je ve firmě v současné době kladen největší důraz, je Webová aplikace Docházka. Na této aplikaci jsem pracoval střídavě po celý čas mé odborné praxe. Aplikace slouží jako komplexní docházkový systém, do kterého lze vkládat docházková data a následně je vyhodnocovat prohlížením vhodně zobrazených vložených dat nebo třeba pomocí přehledných tisknutelných reportů. Aplikace je naprogramována tak, aby byla pro různé zákazníky co nejvíce variabilní a přizpůsobitelná. Ačkoliv je tato aplikace maximálně soběstačná, jsou pro ni vyvinuty další výše popsané klientské Android aplikace, které především zvyšují komfort používání aplikace a také její dostupnost.



Obrázek 3: Aplikace Docházka GIRITON

Aplikace je multitenantní, což znamená, že pro všechny klienty běží jediná aplikace, která podle kontextu zobrazuje klientům data, čímž se znatelně sníží požadavky na výkon hardware, na kterém systém běží. Zároveň se však samozřejmě zvyšuje paměťová náročnost. Při použití této architektury se také musí klást důraz na odstínění jednotlivých klientských účtů od sebe aby nedocházelo k úniku jejich dat.

Aplikace je postavena na nejmodernějších technologiích jakými je i webový framework Vaadin [3] napsaným v jazyce Java [1]. Vaadin je framework, kde na serverové části běží Java. Klientská část pak funguje v klientském prohlížeči pomocí HTML+CSS+JS. Podporovány jsou všechny dnešní prohlížeče nových verzí. HTML, CSS, JS jsou nejnižší úrovní, do které je možné ručně zasahovat, ale framework se o jejich generaci zvládá dostatečně starat sám. Všechny komponenty píše programátor v Javě a ty jsou pak následně pomocí GWT compileru přeloženy právě do JS. Se samotným serverem se pak komunikuje pomocí technologie AJAX [4]. Od této komunikace je však programátor zcela odstíněn.

Grafické komponenty mají zpravidla své serverové a klientské části v jednom. Pro příklad komponenta Button ví, jak se vykreslit na klientovi a zároveň pomocí technologie AJAX komunikuje s částí serverovou. To vše se děje na pozadí, které si řeší framework sám a programátor tak nemusí komunikaci server-client a naopak sám řešit.

```

public class MainForm extends UI
{
    private TextField textFirstname = new TextField();
    private TextField textLastname = new TextField();
    private TextField textEmail = new TextField();
    private Button btnOk = new Button("OK");
    private FormLayout layout = new FormLayout(textFirstname, textLastname, textEmail, btnOk);

    public MainForm()
    {

        btnOk.addClickListener(new ClickListener() {
            @Override
            public void buttonClick(ClickEvent event) {
                //do something with fields ..
                Mail.sendMail("from@from.com", textEmail.getValue(),
                    "User_" + textFirstname.getValue() + "_" + textLastname.getValue() + "_is_
                    the_best.");
            }
        });

        setContent(layout);
    }
}

```

Výpis 2: Vaadin - společný kód pro client a server

Mým úkolem byla implementace či úprava různých formulářů a obrazovek a to pomocí Vaadin frameworku [3]. Většinou šlo o rozšiřování firemní třídy EditorTabVaadin, která je jednou ze základních tříd celé aplikace. S frameworkem jsem se setkal poprvé až ve firmě a proto jsem ho musel pro výkon práce dodatečně nastudovat. Pracoval jsem převážně s různými TextFielddy, Buttony a jinými ovládacími prvky známými z jiných GUI frameworků, které jsem různě podle požadavků umisťoval do různých Vaadin layoutů.

Později jsem pracoval i na doménové části aplikace. Konkrétně se jednalo například o rozšíření docházkového systému o projekty. Tyto projekty zpočátku požadovaly stavební firmy. Nicméně se ukázalo, že jsou prospěšné i v mnoha jiných a jinak zaměřených firmách. Jeden projekt představuje například "Stavba Mostu A" nebo projekt "Reklamace" a od toho se pak může dále odvíjet různá statistika prací na projektu jako je např. strávený čas na projektu.

Data pro doménovou logiku byly persistovány pomocí JPA v podobě ORM frameworku Hibernate [10] do SQL databáze. Ze školy jsem již ovládal základy Hibernate díky volitelného předmětu Java Technologie a tak mi základní práce s tímto frameworkem nečinila větší problémy. Návrh dalších sub-doménových částí probíhal stylem diskuze u tabule a znázornění jednotlivých entit pomocí UML diagramů. Po konečné domluvě došlo k implementaci takové sub-domény.

4.4 JasperReports

Dalším větším úkolem bylo vyřešit jak tvořit přehledné a především tisknutelné reporty. Jednalo se například o různé měsíční výkazy práce nebo rozpisy směn na další měsíce, či výpis služebních cest takovým způsobem, aby značně ulehčili práci třeba při zadávání dat do účetního programu. Pro tyto účely jsme si vybrali knihovnu JasperReports, která zvládá generovat reporty nejen ve formátu pdf, ale umí také například vytvořit dokument tabulkového procesoru MS Excel nebo LibreOffice Calc. Dále zvládá generovat také dokumenty pro software MS Word, či export v podobě HTML stránky.

Každý report se skládá z několika bandů (skupin) jako jsou Page header, Page footer, Detail a dále z různých podskupin, které jdou vytvořit na základě předaných dat reportu. V reportu lze využívat různé grafické elementy jako jsou Labely, TextFieldy, Grafy, Tabulky atd. Data pro report musí být předem připraveny v relační podobě, kde si je pak samotný report může různě seřadit tak, jak zrovna potřebuje. Tyto předlohy pro vygenerování reportu jsou uloženy ve formě XML dokumentu s příponou .jrxml, což zajišťuje to, že si je může upravit klient sám bez nutnosti zásahu firmy. Má práce tedy spočívala v přípravě šablon reportů, jejich dat a spojení se systémem Docházka GIRITON v podobě implementace generování do GUI aplikace.

5 Závěr

Absolvování praxe mi značně rozšířilo mé dovednosti a pomohlo se rozhodnout jakým směrem se dále ubírat. Naučil jsem se pracovat v týmu, což je cenná zkušenost do mého budoucího profesního života.

5.1 Dosavadní znalosti a jejich aplikace

Ve škole jsem se setkal s programovacím jazykem Java, který po mně byl ve firmě požadován v podobě s různými frameworky. Pracoval jsem také s Android frameworkem, se kterým jsem měl možnost pracovat paralelně jak ve firmě, tak ve škole díky volitelnému předmětu Tvorba aplikací pro mobilní zařízení II. Znalosti získané z předmětu jsem využil ve firmě a naopak znalosti z firmy byly zpátky využitelné v předmětu. Dále jsem se setkal s programovacími jazyky SQL(JPQL), které se na škole rovněž vyučují. Zde mi stačily znalosti čistě školní. Dále jsem měl možnost setkat se s jazykem XML, který je ve škole v různých předmětech také probírán. Ve firmě především v kontextu importu/exportu dat z a do jiných systémů. Velmi okrajově pak s HTML a CSS při nutnosti zásahu do vykreslování webového GUI na nejnižší možné úrovni. Užitečnou znalostí mi byla také znalost různých UML diagramů, které byli náplní nejednoho předmětu na škole.

Dále mi při vykonávání praxe pomohli znalosti z těchto předmětů absolvovaných na VŠB:

- Základy programování
- Algoritmy I, II
- Programovací jazyky I, II
- Úvod do softwarového inženýrství
- Úvod do databázových systémů
- Databázové a informační systémy
- Tvorba aplikací pro mobilní zařízení I, II
- Java technologie
- Vývoj informačních systémů

5.2 Nabyté znalosti a zkušenosti během praxe

Pod vedením jsem se naučil jak psát čistější, efektivnější a přehlednější kód, za který jsem nesl zodpovědnost. Špatně napsaný kód mi byl s vysvětlením vrácen k přepracování. Další nabytou zkušeností pro mě byla spolupráce v týmu s čím například souvisí rozdělování a plánování úkolů pomocí systému Trello [13] nebo používání verzovacího systému.

Agilní způsob programování mě taky naučil rozlišovat mezi tím, co je skutečně důležité a na co není potřeba v raných fázích vývoje dávat až takový důraz.

Úplně novým pro mě bylo používání verzovacího systému Apache Subversion [12], který je ve firmě používám pro správu verzí zdrojových kódů. Dnes tento nebo jakýkoliv jiný verzovací systém vidím jako nezbytnou součást týmového vývoje.

Dále jsem se naučil pracovat s frameworkem Vaadin a to do takové míry, že jsem byl schopen úspěšně získat certifikát a stát se tak certifikovaným Vaadin developerem. Důležitou zkušeností pro mě byla práce s Android frameworkem a samotný vývoj Android aplikací, kterému bych se chtěl nadále věnovat. Neméně důležitou zkušeností byla také práce s knihovnou JasperReports.

5.3 Celkové zhodnocení

Možnost absolvování bakalářské práce v podobě praxe mě neuvěřitelně obohatila. Pomohla mi k ověření mých dosavadních znalostí a jejich využití v praxi. Dále jsem se naučil mnoho nových postupů a technologií.

Také jsem se naučil pracovat v několika frameworkcích jakými jsou Vaadin, Android, Hibernate. Dále jsem se naučil pracovat s reportovací knihovnou JasperReports, kterou v budoucnosti určitě využiji.

Důležitým článkem pro mě byla práce v týmu. Ve firmě jsem měl možnost rozdělovat práci, což mě naučilo rozvrhnout a rozdělit práci takovým způsobem, aby bylo dosaženo nejlepších výsledků a to v co nejmenším možném čase. Ve firmě mi také byla nabídnuta další spolupráce, kterou jsem využil.

6 Reference

- [1] SCHILDT Herbert. *Java 7: výukový kurz*. 1. vyd. Brno: Computer Press, 2012, 664 s. ISBN 978-80-251-3748-2.
- [2] The Java™ Tutorials. *An Overview of RMI Applications* [online]. 1995-2015 [cit. 2015-04-12]. Dostupné z: <https://docs.oracle.com/javase/tutorial/rmi/overview.html>
- [3] Vaadin. *Book of Vaadin - vaadin.com* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <https://vaadin.com/book>
- [4] W3Schools. *AJAX Introduction* [online]. 1999-2015 [cit. 2015-04-12]. Dostupné z: http://www.w3schools.com/ajax/ajax_intro.asp
- [5] Android Developers. *Android Developers* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <http://developer.android.com/>
- [6] Android Developers *Managing the Activity Lifecycle* | *Android Developers* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <http://developer.android.com/training/basics/activity-lifecycle/index.html>
- [7] Android Developers. *Gingerbread* | *Android Developers* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <http://developer.android.com/about/versions/android-2.3-highlights.html>
- [8] Android Developers. *Ice Cream Sandwich* | *Android Developers* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <http://developer.android.com/about/versions/android-4.0-highlights.html>
- [9] Android Developers. *Support Library* | *Android Developers* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <http://developer.android.com/tools/support-library/index.html>
- [10] Hibernate. *Hibernate ORM - Hibernate ORM* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <http://hibernate.org/orm/>
- [11] Jaspersoft Community. *JasperReports Library* [online]. 2015 [cit. 2015-04-12]. Dostupné z: <https://community.jaspersoft.com/project/jasperreports-library>
- [12] Apache Subversion. *Subversion* [online]. 2011 [cit. 2015-04-12]. Dostupné z: <https://subversion.apache.org/>
- [13] Trello. *Trello* [online]. 2011 [cit. 2015-04-12]. Dostupné z: <https://trello.com/>
- [14] GIRITON Systems s.r.o.. *Home. Docházka GIRITON* [online]. 2014 [cit. 2015-04-12]. Dostupné z: <http://giriton.cz/cz/index.html#home>
- [15] GIRITON Systems s.r.o.. *Osobní docházka. Docházka GIRITON* [online]. 2014 [cit. 2015-04-12]. Dostupné z: <http://giriton.cz/cz/index.html#section-personal>
- [16] GIRITON Systems s.r.o.. *Píchací hodiny. Docházka GIRITON* [online]. 2014 [cit. 2015-04-12]. Dostupné z: <http://giriton.cz/cz/index.html#section-clock>